Theme Article: Science Gateways: Accelerating Research and Education
Editor: Name, xxxx@email

An EOSC-enabled Data Space Gateway for Climate Science

Donatello Elia
Advanced Scientific Computing Division, Centro Euro-Mediterraneo sui Cambiamenti Climatici, Lecce, Italy

Fabrizio Antonio
Advanced Scientific Computing Division, Centro Euro-Mediterraneo sui Cambiamenti Climatici, Lecce, Italy

Sandro Fiore
Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Paola Nassisi
Advanced Scientific Computing Division, Centro Euro-Mediterraneo sui Cambiamenti Climatici, Lecce, Italy

Giovanni Aloisio
Centro Euro-Mediterraneo sui Cambiamenti Climatici & Università del Salento, Lecce, Italy

***Abstract*—The growing amount of available data in many scientific fields is calling for a radical change in the approach for managing and analyzing these data. The data space concept, a digital ecosystem supporting scientific communities towards a more sustainable and FAIR use of data, has emerged in the last years to address some of the key challenges. This paper presents a domain-specific implementation of the data space concept targeting the needs of climate scientists: the ENES Data Space. Such science gateway has been devised in the context of the European Open Science Cloud to provide climate users with datasets, tools, and services integrated into a single environment for the development of data science applications. The main motivations behind this data space and its architecture are presented in this work, together with an example of scientific application that can be run by users.**

■ **INTRODUCTION** Over the last decade big data has been considered one of the major revolutions in ICT. Besides the most famous three Vs, i.e., volume, velocity, and variety, a fourth one, value, at the core of data economy, has been fostering a data-driven innovation perspective, approach, and thinking, thus gaining strategic importance in the digital market, both for public and private sectors.

The data space concept has been progressively introduced in this context over the last few years, turning out to be suitable also to scientific domains. In particular, the *EU Data Strategy*[1] highlighted the importance of defining a single European data space, a seamless digital area for the development of services based on data.

In that respect, the ongoing efforts in the context of the *European Open Science Cloud* (EOSC) can strongly help the implementation of such concept. As an example, the *EGI-ACE* project[a] is building several data spaces for different scientific communities to support them towards a more sustainable, effective, and FAIR data use.[2]

Among others, the *ENES Data Space* (EDS) is a notable example, relevant to the *European Network for Earth System Modelling (ENES)* community. It has been opened to end users at the end of 2021, offering a single integrated environment with ready-to-use data and programmatic capabilities for the development of data science applications. In particular, this solution integrates into a single environment: (i) Python libraries and frameworks for data analytics and visualization, together with (ii) a large data collection from key community experiments, like the *Coupled Model Intercomparison Project* (CMIP)[b] and the *Coordinated Regional Climate Downscaling Experiment* (CORDEX)[c], and (iii) scalable computing resources that can be deployed on demand on top of the *EGI federated cloud* infrastructure.

Such data space exploits software components from the *Jupyter project*[3] to provide a web-based science gateway available through the EOSC MarketPlace[d] in the context of the EOSC initiative.

A Data Space for Climate Science

The increased models resolution in the development of comprehensive Earth System Models is rapidly leading to very large climate simulations outputs that pose significant challenges in terms of scientific data management, specifically for data sharing, processing, analysis, visualization, preservation, curation, and archiving.[4]

In such domain, large scale global experiments for climate model intercomparison have led to the development of the *Earth System Grid Federation* (ESGF),[5] a federated data infrastructure that involves a large set of modelling centers (data providers) around the globe and includes the European contribution - regarding the ENES community - through the IS-ENES project. Datasets are produced by the different climate modelling institutes, published on their sites, and shared with the whole community through ESGF.

The ESGF infrastructure provides access to climate dataset from various efforts, such as from CMIP. The Coupled Model Intercomparison Project has been established by the Working Group on Coupled Modelling (WGCM)[e] under the World Climate Research Programme (WCRP). CMIP is now in Phase 6 (CMIP6) and its PB-scale database[6] is of strong relevance to WCRP for the IPCC assessments[f].

For more than ten years, ESGF has managed a globally distributed data infrastructure, federating search and replication of datasets among participant institutions. The federation was historically based on nodes, deployed at given sites, divided into more specialised modules: (i) the *Data node*, for publishing and data access, (ii) the *Index node*, for data discovery and metadata indexing, and (iii) the *Identity Provider node*, for authentication and authorization policies. An early approach also included a fourth component, the *Compute node*, which provided some basic analysis features. However, there has always been the interest in extending this architecture to provide more valuable computing services to better support scientists in the climate community in their analysis.

Several efforts have been undertaken over the years; for example in the context of the IS-ENES projects a set of compute services have been established by key European climate research centers. Initial ideas for these computing capabilities are represented by the *analytics-hub concept*[7] and the *ENES Climate Analytics Service* (ECAS) set up in the context of the EOSC initiative by CMCC and DRKZ.[8] ECAS represented an early solution for providing data analysis capabilities close to the ESGF data pools already deployed in the data center, while the analytics-hub focused on providing analytics service on variable-centric

[a] EGI-ACE: https://www.egi.eu/project/egi-ace/

[b] CMIP: https://www.wcrp-climate.org/wgcm-cmip

[c] CORDEX: https://cordex.org

[d] EDS: https://marketplace.eosc-portal.eu/services/enes-data-space

[e] WGCM: https://www.wcrp-climate.org/wgcm-overview

[f] IPCC Reports: https://www.ipcc.ch/reports/

data collections. The ENES Data Space represents an evolution of these early solution targeting a wider user base.

A data space for climate and ESGF would be clearly linked in this landscape, due to the comprehensive data offering from ESGF about datasets of broad interest, from large community experiments. This represents a step forward from the compute service originally envisioned in ESGF towards a more integrated data management and analysis solution.

From an architectural perspective, a climate data space and ESGF would work at two different levels: on one hand, ESGF would mainly operate a federated data archive through a set of data nodes distributed worldwide across different modelling centers; on the other hand, a climate data space would provide a single entry-point to compute capabilities co-located with a local data store hosting a specific data selection from ESGF.

As already mentioned, the data offering is at the core of a climate data space. However, it is important to remark that data alone would not be enough without a comprehensive ecosystem, which also includes a potentially wide set of tools and services enabling scientists to: develop applications, analyze data, visualize results, understand provenance, reproduce analyses, share code and document workflows, and publish new products. Data, services, and tools represent the three fundamental components of a climate data space ecosystem.

A Science Gateway for the ENES Community

The EDS environment proposed in this work, strives to provide climate scientists with a ready-to-use research environment accessible through a web-based science gateway. From a technical standpoint, EDS is a specific implementation of a climate data space developed in the context of the EU-funded project EGI-ACE. It provides a complete data science environment enabling users to perform scientific analysis on large climate datasets. It integrates computing and storage resources, climate datasets as well as various data access, analytics and visualization modules, tools, and parallel computing frameworks.

The development of such data space has been driven by a set of key needs and requirements identified through the experience gained by working with climate scientist.

The typical methodology which has been used by scientists in the past consisted in downloading the climate datasets from central repositories (e.g., ESGF) on their desktop machines and then processing data locally with home-made scripts. The increasing volumes of data made this approach no longer viable due to long download time and limited computing/storage resources on local machines. Providing scientists with large ready-to-use datasets and software solutions *closely integrated* with the computing infrastructure can almost completely eliminate the setup time and dramatically increase their *productivity*.

This integration with Cloud/HPC infrastructures enables the use of High Performance Data Analytics (HPDA) solutions that can easily scale the analysis on thousands of computing units supporting *parallel processing*, which has become fundamental for effectively addressing the increasing scale in climate data.

From a functional point of view, *interactive*, *exploratory* and *batch* (i.e., workflow-based) analysis are all very common methodologies used by scientists for investigating and analyzing data. Providing support for these different methodologies trough a *single gateway*, while hiding from the infrastructural details, will allow to fully address user needs in terms of data access, processing and visualization.

Another relevant aspect is *research outputs sharing*, i.e., data products and application definition. This is a key aspect towards pursuing the goals of *Open Science* and increasing the collaboration between different scientists and groups.

As an overarching goal, the data space can *democratize data analysis* by providing opportunities to new user groups and communities who do not currently have access to computing infrastructures capable of supporting such large-scale analysis.

Data Space Architecture

The EDS architecture has been defined starting from the identified needs. Figure 1 shows a high-level view of the data space architecture, highlighting the main infrastructural components considered.

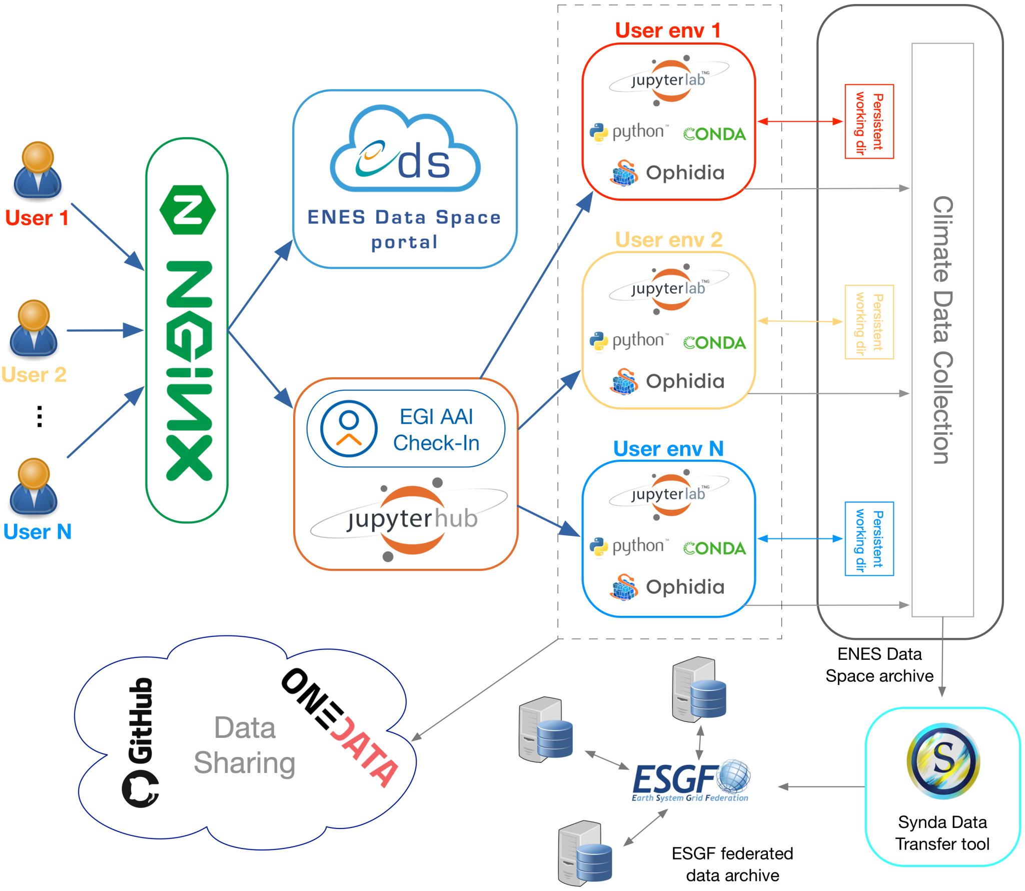


Figure 1. High-level architecture of the data space. The main services and tools integrated in the software infrastructure are highlighted

Data Science Gateway Interface and AAI

The science gateway of the EDS is implemented through multiple services accessible via a Ngnix-based proxy, as it can be observed in the left-most part of Figure 1. These services include the EDS web portal[g] and an instance of *JupyterHub*,[3] a service for running a multi-user Jupyter server environment.

AAI aspects to the platform are handled through the *EGI Check-in*[h] service, directly integrated into JupyterHub for managing user authentication and authorization. EGI Check-in provides users with a uniform, easy and secure way for accessing services, also supporting authentication through institutional and social media accounts. It enables federated authentication and authorization of users for dealing with identity management and access control aspects (e.g., definition of virtual organizations).

Once the user is authenticated, JupyterHub server spawns a new instance of single-user environment with the enhanced *JupyterLab* server: the latest web-based user interface for Project Jupyter.

Data Science Environment The JupyterLab server is linked to a pre-configured ecosystem of Python-based tools and frameworks for parallel data analysis and result visualization. This environment provides the features for the execution of scientific code in the Jupyter Notebooks.

The environment, based on the popular *Conda* package manager, contains a wide set of pre-configured scientific modules from the Python ecosystem targeting the climate commu-

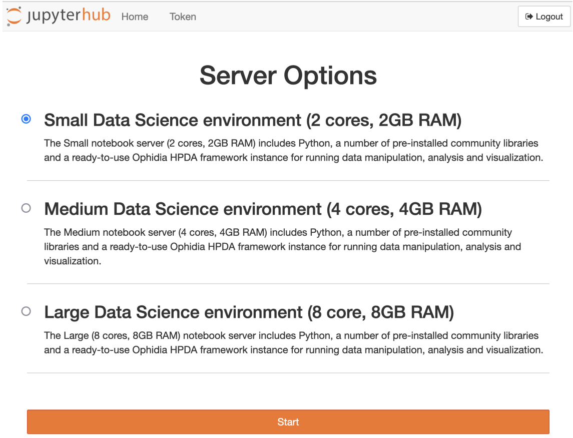
[g] EDS Portal: https://enesdataspace.vm.fedcloud.eu/

[h] EGI Check-in: https://www.egi.eu/service/check-in/

nity needs, such as *Xarray*,[9] *Intake*[i], as well as computing frameworks for parallel analysis like *Dask*[10] and *Ophidia*.[11] This is displayed in the central part of Figure 1.

The whole JupyterLab server and the Python ecosystem is implemented through a Docker container for easy deployment on the underlying cloud platform. In this way, multiple environments with different tools can be easily created and added to the infrastructure.

In order to support different application requirements, the science gateway, allows users to select the computing resources (i.e., in terms of CPUs and RAM) required for their needs. Different computational profiles have been defined according to the most frequent scenarios requested by scientists. The user environment is deployed on-demand on the cloud infrastructure according to the profile selected at login time. Figure 2 shows the possible profiles currently made available. The set of profiles is periodically updated to take into account new user requirements and applications workload.



Server Options

● Small Data Science environment (2 cores, 2GB RAM)

The Small notebook server (2 cores, 2GB RAM) includes Python, a number of pre-installed community libraries and a ready-to-use Ophidia HPDA framework instance for running data manipulation, analysis and visualization.

○ Medium Data Science environment (4 cores, 4GB RAM)

The Medium notebook server (4 cores, 4GB RAM) includes Python, a number of pre-installed community libraries and a ready-to-use Ophidia HPDA framework instance for running data manipulation, analysis and visualization.

○ Large Data Science environment (8 core, 8GB RAM)

The Large (8 cores, 8GB RAM) notebook server includes Python, a number of pre-installed community libraries and a ready-to-use Ophidia HPDA framework instance for running data manipulation, analysis and visualization.

Figure 2. On-demand provisioning of user-based environment on the cloud. Profiles with different computing requirements are available

Cloud-based Computing Platform

The set of services and user environments provided by the science gateway are deployed on a a state-of-the-art cloud infrastructure, which provides the scalable computing platform for data and compute-driven applications. As previously mentioned, each service is handled as a Docker container managed by a Kubernetes cluster.

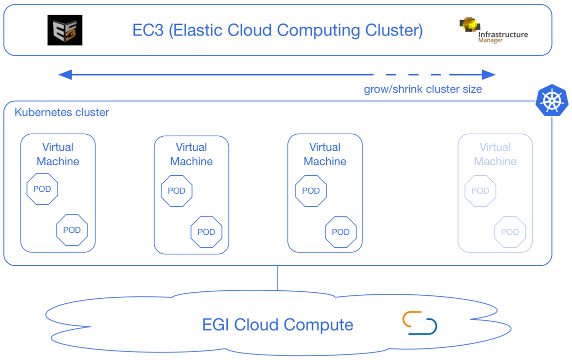


Figure 3. Kubernetes cluster for the EDS infrastructure on top of the EGI FedCloud

The actual resources for the execution of the container-based services are provided by the EGI Federated Cloud e-Infrastructure[j]. Figure 3 shows a high-level view of the deployment of the data space infrastructure on top of the physical cloud resources. More specifically, the *EC3 - Elastic Cloud Computing Cluster* - service[12] is used to manage and elastically adapt the deployment size of a Kubernetes cluster. In this way, the resources assigned to the data space (i.e., the number of virtual machines) dynamically grow and shrink according to the actual users' workload, measured, for example, in terms of memory usage, CPUs and number of Kubernetes pods (i.e., the smallest unit of application that can be deployed in Kubernetes) in a virtual machine.

Kubernetes[k] is used to automate the deployment of the Docker-based micro-services and single-user environments of the EDS system. In this respect, Kubernetes allows managing the deployment of consistent and isolated environments with different resource configurations on top of the cloud infrastructure. In particular, the *JupyterHub Kubernetes Spawner* (also known as KubeSpawner)[l] is in charge of spawning across the Kubernetes cluster nodes and managing the environment containers once a user logs into the system.

[i] Intake: https://intake.readthedocs.io/en/latest/index.html

[j] EGI FedCloud: https://www.egi.eu/service/cloud-compute/

[k] Kubernetes: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

[l] kubespawner: https://jupyterhub-kubespawner.readthedocs.io

Data Access and Sharing The infrastructure is linked to 150TB of storage resources for the EDS file system. Each user is provided with a persistent dedicated storage space on the file system, which is exposed to the various Kubernetes virtual machines composing the cluster and mapped into the user container at deployment time. This storage space can be used by scientists to store their notebooks and results between their experiment sessions, prior to publication and sharing with other users.

Moreover, each user has access to a large collection of ready-to-use datasets from the ESGF federated archive, curated and stored on the system. The data collector is based on *Synda*[m], which allows datasets downloading and (one-way) synchronization of the local data pool with the data hosted on the ESGF data infrastructure.

Results, output products and the experiment definitions (in the form of Jupyter Notebooks) can be easily shared among users through the data sharing services included in the environment, i.e,. *Onedata*[n]. Notebooks can also be published and shared on GitHub and accessed within the data space thanks to the integration of the related JupyterLab extension.

Supported Scientific Applications

The data space enables various kinds of usages for data-centric research scenarios in the climate science domain supporting, for example, interactive analysis, parallel processing and workflows composed of several analytics tasks.

This section presents a typical interactive analysis that can be implemented by a climate scientist for the computation of a climate change indicator: the *tropical nights index* (i.e., the number of days where the daily minimum temperature is above a given threshold)[o]. Figure 4 shows some of the stages of the analysis implemented in the form of Jupyter Notebook (in Python) through the JupyterLab computing environment provided by the science gateway. It is noteworthy that these stages could be generalized for different types of analytics applications.

Stage 1: Data Search & Discovery The first step of the analysis starts with the selection of the input datasets. The EDS is equipped with a set of ready-to-use specific *CMIP* and *CORDEX* variable-centric collections from the ESGF federated data archive. Additional data can be integrated into the collection based on users' needs, who can ask for new datasets by filling out a request form available on the data space portal. Synda will take care of retrieving the requested data and keeping the data space synchronized with the ESGF archive.

Since the set of available data is quite large, before analyzing it, users need to understand which datasets are available, the metadata describing each dataset, and how they can select and access a specific dataset. In this regard, users can exploit the *intake-esm* data cataloging utility integrated in the environment. This can be invoked directly within a notebook to parse and execute queries against the Earth System Model catalog associated with the EDS archive, thus selecting in a simple way specific datasets from those available in the collection. Figure 4.a shows how the intake module can be used to explore the set of available data and retrieve those relevant for the analysis. In this particular case the *daily minimum temperature (tasmin)* is used for the computation of the indicator.

The data space allows to improve scientists productivity by providing a set of curated data immediately usable without preliminary download, together with the tools for querying and identifying the data required for their analysis.

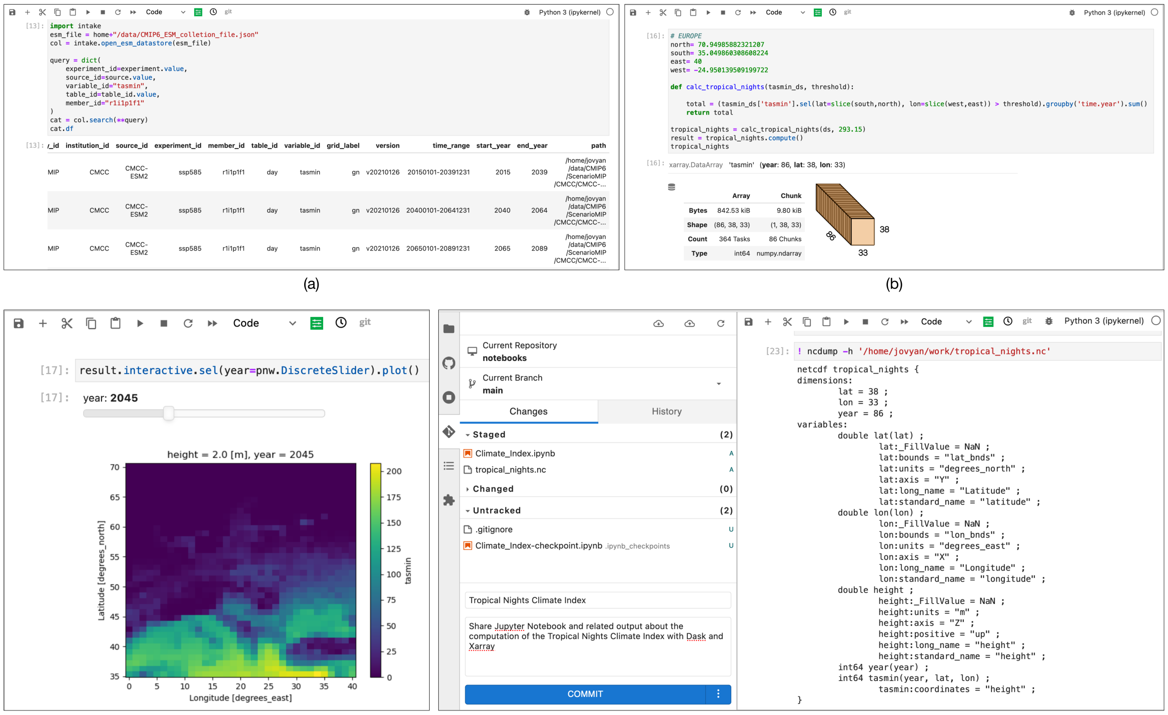
Stage 2: Parallel Data Analysis Once the data have been identified by querying the system, intake-esm can also be used for loading data assets (i.e., NetCDF files), for example, into Xarray datasets to run the analysis. Alternately, scientists can exploit the Ophidia framework and its Python bindings, *PyOphidia*, to perform data analysis on multidimensional scientific datasets. These two solutions integrated in the data space computing environment also allow parallel data processing. Moreover, Ophidia supports batch execution of complex analytics workflows.

In this particular case, the analysis is based on Xarray jointly used with Dask to run in parallel the computation of the tropical nights on

[m] Synda: https://is.enes.org/sdm-sync-data/

[n] OneData https://onedata.org/

[o] List of ETCCDI Indices: http://etccdi.pacificclimate.org/list_27_indices.shtml



```
[13]: import intake
      esm_file = home+"/data/CMIP6_ESM_colletion_file.json"
      col = intake.open_esm_datastore(esm_file)

      query = dict(
          experiment_id=experiment.value,
          source_id=source.value,
          variable_id="tasmin",
          table_id=table_id.value,
          member_id="r1i1p1f1"
      )
      cat = col.search(**query)
      cat.df
```

[13]:

_id	institution_id	source_id	experiment_id	member_id	table_id	variable_id	grid_label	version	time_range	start_year	end_year	path
MIP	CMCC	CMCC-ESM2	ssp585	r1i1p1f1	day	tasmin	gn	v20210126	20150101-20391231	2015	2039	/home/jovyan /data/CMIP6 /ScenarioMIP /CMCC/CMCC-...
MIP	CMCC	CMCC-ESM2	ssp585	r1i1p1f1	day	tasmin	gn	v20210126	20400101-20641231	2040	2064	/home/jovyan /data/CMIP6 /ScenarioMIP /CMCC/CMCC-...
MIP	CMCC	CMCC-ESM2	ssp585	r1i1p1f1	day	tasmin	gn	v20210126	20650101-20891231	2065	2089	/home/jovyan /data/CMIP6 /ScenarioMIP /CMCC/CMCC-...

(a)

```
[16]: # EUROPE
      north= 70.94985882321207
      south= 35.049860308608224
      east= 40
      west= -24.950139509199722

      def calc_tropical_nights(tasmin_ds, threshold):

          total = (tasmin_ds['tasmin'].sel(lat=slice(south,north), lon=slice(west,east)) > threshold).groupby('time.year').sum()
          return total

      tropical_nights = calc_tropical_nights(ds, 293.15)
      result = tropical_nights.compute()
      tropical_nights
```

[16]: xarray.DataArray 'tasmin' (**year**: 86, **lat**: 38, **lon**: 33)

	Array	Chunk
Bytes	842.53 kiB	9.80 kiB
Shape	(86, 38, 33)	(1, 38, 33)
Count	364 Tasks	86 Chunks
Type	int64	numpy.ndarray

(b)

```
[17]: result.interactive.sel(year=pnw.DiscreteSlider).plot()
```

[17]: year: **2045**

(c)

Tropical Nights Climate Index

Share Jupyter Notebook and related output about the computation of the Tropical Nights Climate Index with Dask and Xarray

```
[23]: ! ncdump -h '/home/jovyan/work/tropical_nights.nc'
netcdf tropical_nights {
dimensions:
        lat = 38 ;
        lon = 33 ;
        year = 86 ;
variables:
        double lat(lat) ;
                lat:_FillValue = NaN ;
                lat:bounds = "lat_bnds" ;
                lat:units = "degrees_north" ;
                lat:axis = "Y" ;
                lat:long_name = "Latitude" ;
                lat:standard_name = "latitude" ;
        double lon(lon) ;
                lon:_FillValue = NaN ;
                lon:bounds = "lon_bnds" ;
                lon:units = "degrees_east" ;
                lon:axis = "X" ;
                lon:long_name = "Longitude" ;
                lon:standard_name = "longitude" ;
        double height ;
                height:_FillValue = NaN ;
                height:units = "m" ;
                height:axis = "Z" ;
                height:positive = "up" ;
                height:long_name = "height" ;
                height:standard_name = "height" ;
        int64 year(year) ;
        int64 tasmin(year, lat, lon) ;
                tasmin:coordinates = "height" ;
}
```

(d)

Figure 4. Different stages of a climate index computation from the EDS JupyterLab interface. Clock-wise from top left: a) querying of the data catalog, b) execution of the analysis, c) results visualization, d) sharing of the results

the whole data set composed of 86 years from 2015 to 2100. The computation is performed by using all the resources made available by the Kubernets Pod allocated by the user: in this case the ”medium” profile image with 4 computing cores and 4GB of memory has been used. Figure 4.b shows the code used for the computation as well as a graphical representation of the resulting data structure provided by Xarray.

As it can be argued, one of the main benefits provided by the science gateway is that it allows to easily adapt the environment requirements based on the applications needs and scale the analysis on multiple computing units. Thanks to the gateway and pre-configured environment, users do not need computer scientist skills to set and scale up their applications.

Stage 3: Data Visualization Visualization of results represents the following step in the pipeline for understanding the results of the analysis. The data space software stack includes a wide set of libraries to perform data visualization. In this regard, users can exploit the Xarray’s plotting capabilities to easily create informative plots. On top of this, they can directly use *Matplotlib* and *Cartopy* libraries to create highly-customized and publication-quality maps and geospatial visualizations, as well as other solutions like *hvPlot*, *Bokeh* or *Plotly* for interactive plots.

Figure 4.c shows the tropical nights index map generated starting from the results of the analysis. The *hvPlot* module integrated in Xarray allows to easily create a slideshow of the indicator’s different time steps.

Through the integration of multiple libraries, the Jupyter Notebook-based implementation allows to easily tailor and customize the visualization and better highlight the results of the analysis, leading to publication-ready figures. It is important to mention that users can also manually add other libraries to their account for specific needs.

Stage 4: Results Reuse & Sharing As introduced earlier, collaboration between research groups is key to future scientific discovery in lieu of the Open Science paradigm. In this respect, the reuse of analysis workflows and the possibility to share code and results are critical aspects. Jupyter Notebooks represent an easy way to create and share computational documents embedding code, explanation and results, thus supporting code reuse as well. This could also be a valuable aspect for training and education purposes in order to improve productivity through knowledge sharing and to address recurring learning difficulties.

Through the integration with different supporting tools, the EDS offers users the possibility to publish and share their experiments and analysis results and to make them available to different user groups. Figure 4.d shows, for example, how the notebook for the tropical night computation and the resulting NetCDF file can be uploaded on the user's GitHub repository directly from the JupyterLab interface (through the proper extension) without needing to download data locally first. This allows for an increased productivity since the whole scientific workflow is supported by a single integrated environment.

Conclusions

This article presented the EDS, an EOSC-based data science environment for the climate community developed in the context of the EGI-ACE project.

The main motivations behind this data science environment have been introduced, together with the key requirements addressed, the software architecture devised to enable interactive and parallel data analysis/visualization, the supported scientific applications as well as the interactive gateway based on Jupyter Notebooks.

The environment was made accessible to end users in 2021, and it has been periodically updated with new software releases, datasets, and tools. This will continue as a natural extension of the data space, especially driven by the users' feedback and requests. Current ongoing developments take into account the use of accelerators (e.g., GPUs) to support Artificial Intelligence-based applications, as well as solutions for distributing the processing in a cloud environment across multiple virtual machines.

The integration of these data science environments with High Performance Computing (HPC) infrastructures still represents an open challenge with respect to supporting large-scale, cloud-enabled analytics applications. The production-level integration of such capabilities into HPC ecosystems represents future work for the EDS, which will enable transparent provisioning of containerized data analytics solutions through novel *HPC as a Service* (HPCaaS) paradigms.

Acknowledgments

This work was supported in part by the EGI-ACE project (Horizon 2020) under Grant number 101017567 and by the EU H2020 IS-ENES Phase 3 (ISENES3, Grant Agreement 824084). We thank Enol Fernández from EGI Foundation, Miguel Caballer from Universitat Politècnica de València, Hakan Bayindir from TUBİTAK ULAKBİM and the Synda team from Institut Pierre Simon Laplace (IPSL) for their tecnhical support. Moreover, we thank Antonio Aloisio from CMCC for his editing and proof-reading work on this article.

REFERENCES

1. "A European strategy for data - communication from the commission to the european parliament, the council, the european economic and social committee and the committee of the regions," European Commission, Tech. Rep., 2 2020. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52020DC0066&qid=1619802547376
2. M. D. Wilkinson *et al.*, "The fair guiding principles for scientific data management and stewardship," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016. [Online]. Available: https://doi.org/10.1038/sdata.2016.18
3. B. E. Granger and F. Pérez, "Jupyter: Thinking and storytelling with code and data," *Computing in Science Engineering*, vol. 23, no. 2, pp. 7–14, 2021. [Online]. Available: https://doi.org/10.1109/MCSE.2021.3059263
4. J. L. Schnase *et al.*, "Big data challenges in climate science: Improving the next-generation cyberinfrastructure," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 3, pp. 10–22, 2016. [Online]. Available: https://doi.org/10.1109/MGRS.2015.2514192
5. L. Cinquini *et al.*, "The earth system grid federation: An open infrastructure for access to distributed geospatial data," *Future Generation Computer Systems*, vol. 36, pp. 400–417, 2014. [Online]. Available: https://doi.org/10.1016/j.future.2013.07.002

6. V. Balaji *et al.*, “Requirements for a global data infrastructure in support of CMIP6,” *Geoscientific Model Development*, vol. 11, no. 9, pp. 3659–3680, 2018. [Online]. Available: https://doi.org/10.5194/gmd-11-3659-2018
7. S. Fiore *et al.*, “Towards an open (data) science analytics-hub for reproducible multi-model climate analysis at scale,” in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 3226–3234. [Online]. Available: https://doi.org/10.1109/BigData.2018.8622205
8. S. Bendoukha *et al.*, “Enabling server-based computing and fair data sharing with the enes climate analytics service,” in *2019 15th International Conference on eScience (eScience)*, 2019, pp. 651–653. [Online]. Available: https://doi.org/10.1109/eScience.2019.00103
9. S. Hoyer and J. Hamman, “xarray: N-D labeled arrays and datasets in python,” *Journal of Open Research Software*, vol. 5, no. 1, 2017. [Online]. Available: http://doi.org/10.5334/jors.148
10. M. Rocklin, “Dask: Parallel computation with blocked algorithms and task scheduling,” in *Proceedings of the 14th Python in Science Conference*, K. Huff and J. Bergstra, Eds. Austin, TX, USA: SciPy, 2015, pp. 130 – 136. [Online]. Available: https://doi.org/10.25080/Majora-7b98e3ed-013
11. D. Elia *et al.*, “Towards HPC and big data analytics convergence: Design and experimental evaluation of a HPDA framework for eScience at scale,” *IEEE Access*, vol. 9, pp. 73 307–73 326, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3079139
12. M. Caballer *et al.*, “Ec3: Elastic cloud computing cluster,” *Journal of Computer and System Sciences*, vol. 79, no. 8, pp. 1341–1351, 2013. [Online]. Available: https://doi.org/10.1016/j.jcss.2013.06.005

Donatello Elia is a junior data scientist at the Advanced Scientific Computing (ASC) Division, Euro-Mediterranean Centre on Climate Change (CMCC) Foundation, Italy. His main research interests include data science, HPC, cloud computing, big data, and scientific data management. He received the Ph.D. degree in engineering of complex systems from the University of Salento, Italy, in 2021. He is a member of the IEEE and the IEEE Computer Society. Contact him at donatello.elia@cmcc.it.

Fabrizio Antonio is a research engineer at the Advanced Scientific Computing (ASC) Division, Euro-Mediterranean Centre on Climate Change (CMCC) Foundation, Italy. His research activities focus on scientific data management, high-performance data analytics, cloud computing and container orchestration. He was awarded a Master’s Degree with first-class honors in Computer Engineering from the University of Salento, Faculty of Engineering, in April 2016. Contact him at fabrizio.antonio@cmcc.it.

Sandro Fiore is an Associate Professor at the Department of Information Engineering and Computer Science (DISI), University of Trento, where he also leads the High Performance Climate Informatics Laboratory and a Lecturer at the School of Innovation. His research interests include data science and learning, scientific data management, big data, and artificial intelligence applied to climate change in extreme-scale HPC environments. He is a member of IEEE and ACM. Contact him at sandro.fiore@unitn.it.

Paola Nassisi is a junior data scientist and deputy director at the Advanced Scientific Computing (ASC) Division, Euro-Mediterranean Centre on Climate Change (CMCC) Foundation, Italy. Her activities focus on big data, IoT and scientific data analysis, and visualization in the context of climate change. She received a Master Degree cum laude in computer engineering from the University of Salento, Italy, in 2011. Contact her at paola.nassisi.cmcc.it.

Giovanni Aloisio is Professor Emeritus of Information Processing Systems at the Department of Innovation Engineering of the University of Salento, Lecce-Italy and Director of the Euro-Mediterranean Centre on Climate Change (CMCC) Supercomputing Center. He is member of the CMCC Strategic Council and the ENES HPC Task Force. He has contributed to the IESP and EESI exascale roadmaps. He is the author of more than 100 papers in referred journals on HPC, grid & cloud computing and distributed data management. Contact him at giovanni.aloisio@cmcc.it.